unit value, the DATA_ID key also includes text to be printed with the raw data indicating the units of the value, such as "Mhz," "ns," "hex," etc.

Encoded Value: 0xFFF0

Format

The format for the DATA_ID descriptor key includes a field title describing the data, length of the data in bytes, format of the data, text to be printed with the data, and a field description.

```
dw    DATA_ID
db    Null terminated string for field title
db    Length of data field in bytes
db    Format ID
db    Null terminated string which follows the printed numeric data value
db    Null terminated string for field description
```

Notes:

The Format ID is a special indicator flag that specifies how the data is to be printed. It may also be used in cases where the data needs to have some formula applied to it to derive a meaningful value to display to the user. If a formula is used, the formula is typically coded in the utility program. Table 5 lists the current codes for the format of the data. It must be understood that additional formats could be created depending on the type of raw data to be retrieved and displayed.

Table 5

Format Code	Encoded Value	Notes
FORMAT_SEE_DETAILS	0	Prints a message for the user to see a details window for more information
FORMAT_HEX	1	Prints the data in hexadecimal
FORMAT_DECIMAL	2	Prints the data in decimal
FORMAT_BIOS_ROMSIZE	3	Prints the data according to the formula 64k*(n+1).
FORMAT_2_TO_N	4	Prints the data according to the formula 2^n. Used for memory size.
FORMAT_EXTENDED_DATA	5	Format used for extra data that may exist which has no specific (or known) definition. An example would be any undefined extension bytes at the end of Type 0 or the variable SMBIOS specification). Extended data is generally used

Table 5		
Format Code	**Encoded Value**	**Notes**
		within a repeat loop, therefore, when encountering this key type the utility program prints "Extended Value Data Number:" followed by the current iteration of the loop.
FORMAT_MEMORY_CAPACITY	6	Prints "Unknown Capacity" the value is 8000000h, or the value in kB.

Assembler Example:

Below is an example of the Field definition for the speed of the external clock, which is Type 4, offset 12h in the SMBIOS database.

```
dw    DATA_ID                                          ; Data identifier
db    "External Clock",0                               ; Field description
db    02h                                              ; Length of data in bytes
db    FORMAT_DECIMAL                                   ; Printed data format
db    "Mhz",0                                          ; Any text to follow data
db    "Clock frequency of the external clock (in Mhz)",0  ; Field description
```

B. STRING_ID

The STRING_ID descriptor key is used for fields that consist of strings and informs the utility program that this offset represents the number of a string stored in the SMBIOS database.

Encoded Value: 0xFFF1

Format

The format for the STRING_ID descriptor key includes a field title describing the data, text to be printed with the data, and field description.

```
dw    STRING_ID
db    Null terminated string for field title
db    Null terminated string for field description
```

Assembler Example:

Below is an example of the structure definition for the vendor name of the BIOS software used in the computing system, which is Type 0, offset 04h in the SMBIOS database.

```
dw    STRING_ID                                  ; String identifier
db    "BIOS Vendor's Name",0                     ; Field description
db    "String indicating the name of the BIOS vendor",0  ; Field description
```

C. BIT_FIELD_ID and ENDOF_BIT_FIELD_ID

These descriptor keys define a bit field. A bit field is a data value whose individual bits represent one of two possible states. The BIT_FIELD_ID key begins description of each bit in the field and provides appropriate text to be printed based on whether the individual bits are 1 or 0. The ENDOF_BIT_FIELD_ID key indicates the end of the bit definitions, so that not all bits within the bit field have to be defined in the template file. For example, if the byte size is 2, but bits 11-15 are reserved or undefined, their definitions can be left out of the template file.

Encoded Value:

BIT_FIELD_ID	0xFFF2
ENDOF_BIT_FIELD_ID	0xFFF3

Format

The format for the BIT_FIELD_ID descriptor key includes a field title describing the data, size in bytes of the bit field data value, text to display based on whether the bits are a 1 or a 0, the number of each bit in the bit field and a description for each bit.

```
dw    BIT_FIELD_ID
db    Null terminated string for field title
db    Size in bytes of the bit field data value
db    Null terminated string to use when bit = 0
db    Null terminated string to use when bit = 1
db    xx - Bit Field bit number
db    Null terminated string describing the previous bit number
. . . repeat this section until all bits are defined . . .
dw    ENDOF_BIT_FIELD_ID
```

Assembler Example:

Below is an example of use of these keys to create a structure definition for the bit field stored in the SMBIOS database describing the error correcting capabilities of the memory controller of the computing system. This information is Type 5, offset 05h in the SMBIOS database. The bit field for this information consists of six (6) bits, where